

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

Programų sistemų kūrimo darbas

**TSP ataskaita**  
(TSP report)

Atliko: 3 kurso 3 grupės studentai

Mėnuliukai (parašas)

Darbo vadovas:

Doc., Dr. Saulius Ragaišis (parašas)

Vilnius  
2019

## Turinys

Ivadas .....	2
1. Startavimas .....	2
1.1. Rolės .....	2
1.2. Komandos tikslai .....	2
1.3. Asmeniniai tikslai .....	2
1.4. Komandos susitikimai .....	2
1.5. Reikalavimai duomenims .....	3
2. Strategija .....	3
3. Planavimas .....	3
4. Reikalavimai .....	3
5. Projektavimas .....	3
6. Įgyvendinimas .....	4
7. Testavimas ir integravimas .....	4
8. Aptarimas .....	4

# **Įvadas**

Šioje ataskaitoje pateikiame, kaip mūsų komandai („Mėnuliukai“) sekėsi taikyti TSP principus, kurie principai buvai taikomi ir kaip jie įtakoją komandos produktyvumą ir padarytus darbus.

## **1. Startavimas**

### **1.1. Rolės**

Dar prieš gaunant reikalavimus iš užsakovo buvo pasiskirstyta rolėmis. Rolės buvo paskirstytos pagal tai, ką komandos narys moka geriausiai.

- Matas Savickis - komandos lyderis
- Andrius Bentkus - backend specialistas
- Justas Tvarijonas - frontend specialistas
- Džiugas Mažulis - programuotojas
- Greta Pyrantaitė - programuotoja

### **1.2. Komandos tikslai**

1. Kaip galima greičiau sukurti mažiausią užskaitomą produktą(MVP)
2. Likusį laiką po MVP paskirti papildomiems reikalavimams įgyvendinti ir tobulinti produktą
3. Baigti darbą likus mėseniui iki atsiskaitymo datos

### **1.3. Asmeniniai tikslai**

1. Išmokti React technologiją
2. Įsisavinti Agile darbo komandoje metodologiją
3. Prisiimti asmeninę atsamokybę už produkto kokybę. Pralaimiu ne aš, pralaimi visa komanda

### **1.4. Komandos susitikimai**

Komanda nutarė dėl vienos dienos savaitėje, kada visa komanda susitiks aptarti praeitos savaitės rezultatus ir suplanuoti ateinančios savaitės darbus. Susitikimai vykdavo laikantis šių žingsnių:

1. Aptariame praeitos savaitės darbus, kas buvo gerai ir blogai
2. Aptariame kokius darbus reiks padaryti ateinančią savaitę

3. Vyksta balsavimas, kiek valandų užtruks padaryti kiekvieną atskirą užduotį(Agile poker)
4. Iš nubalsuotų užduočių sąrašo paimama 40 valandų vertės užduočių ir jos pridedamos į ateinančios savaitės užduočių sąrašą
5. Jeigu yra užduočių, kurias reikia padaryti kuo greičiau, tos užduotys būna priskiriamos savanoriams

## **1.5. Reikalavimai duomenims**

Savaitės gale būna sekama, kiek užduočių liko nepadaryta arba kiek buvo padaryta daugiau nei tikėtasi, ir pagal valandų skaičių vertinama, ar keisti ateinančios savaitės vertinimus, mažinti valandas, didinti valandas ar kitaip koreguoti darbą, kad būtų padaroma kuo daugiau.

## **2. Strategija**

Paskaičius ir įvertinus reikalavimus buvo nutarta, kad darbui atlikti reikės 2 mėnesių(1 ciklas - 1 savaitė) ir kiek dar liks laiko iki atsiskaitymo projekto tobulinimui. Buvo nutarta naudotis Java Spring ir Javascript React.js technologijomis. Technologijos buvo pasirinktos pagal jų populiarumą, norą jas išmokti(tiems komandos nariams, kurie nebuvo susidūrę su šiomis technologijomis) ir technologijų galią siekiant greitai kurti geros kokybės produktą.

## **3. Planavimas**

Užduotis dažniausiai surašo komandos lyderis pasitaręs su frontend ir backend specialistais komandoje. Pagal programų sistemų kūrimo(PSK) dalyko sandaros aprašymą ir kiek aprašyme numatoma individualaus darbo valandų buvo nutarta(preliminariai paskaičiuota), kad vienas žmogus per savaitę turi padaryti 8 valandas individualaus darbo, todėl vienam ciklui(1 savaitei) planuojama 40 valandų darbo

## **4. Reikalavimai**

Pagal užsakovo pateiktus prašymus ir atsakymus į klausimus, kuriuos jam pateikėme, buvo sudarytas dokumentas, išvardinantis visus reikalavimus, kuriuos reikės įgyvendinti. Iš reikalavimų buvo išgauta 14 istorijų(stories), kurios bus mūsų komandos priėmimo testavimo sąrašas.

## **5. Projektavimas**

Sistema buvo projektuojama pagal MVC šabloną, kuriame kiekviena technologija turėjo aiškiai paskirtį(M ir C - Java Spring, V - React.js). React.js mums suteikė komponentinio programavimo

privalumus(UI komponentų perpanaudojimas), Spring mums suteikė greitą būdą kurti ir testuoti backend'ą.

## **6. Įgyvendinimas**

Siekiant sumažinti blogą kodą ir klaidas kodo bazėje buvo įvesta taisyklė, kad bent vienas kitas komandos narys turi peržiūrėti, pasileisti ir įsitikinti, kad kodas veikia teisingai prieš tą kodą pridėdant į bendrą kodo bazę.

## **7. Testavimas ir integravimas**

Formalūs testai nebuvo vykdomi, tik patikrinama, ar kodas veikia iš pirmo žvilgsnio. Surastos klaidos būdavo užregistruojamos į užduočių planavimo platformą Trello ir pagal klaidos sunkumą būna įtraukiamos arba neįtraukiamos į sekančią savaitę.

## **8. Aptarimas**

Aptarimas būna vykdomas neformaliai kiekvieną savaitę planavimo metu. Pasikalbama, kas buvo gerai ar negerai, pasižadama daugiau nebedaryti blogų dalykų ir dirbti, nors vistiek nedarbant sekantį ciklą.